Amendments to the Specification:

Please replace paragraphs [0024], [0026], [0034], [0036] and [0038] with the following amended paragraphs:

[0024]   As is conventional in turbo decoders, three quantities, alpha (α), beta (β) and gamma (γ) are defined. For a specific state and a specific time step, α has a value that defines the probability that the coder is at that state at that specific time step. Alpha is derived recursively starting from time k=1 and moving forward in time. The value β is similar to α but works backwards in time. Gamma (γ) is defined as the transition probability that the coder will move ~~from~~ from a state at a given time to some allowed state at the next subsequent time increment. Alpha (α) can be calculated for all states in a trellis based on the state transition probabilities represented by gamma (γ). The gamma (γ) calculation performed at stage 18 is stored in register 22. Calculation stages 24 and 26 respectively calculate each alpha and normalize the alpha calculations. Each alpha (α) value is calculated based on the input from register 22 as well as the previously calculated alpha value provided at input 24b and outputted from calculation stage 26 through multiplexer 28 and register 30, which holds eight (8) calculated values. The output of register 30 is coupled to the input of alpha memory 32 which stores the first calculated alpha value at the first memory location ~~23a~~ 32a and also provides the calculated alpha value to input 24b.

[0026]   Initially, all of the alpha values are calculated, whereupon the beta values are calculated by utilization of the sp1p2 data which ~~are~~ read in reverse order from local memory 20 (i.e., "last-in, first-out order") in order to perform the calculations required for the backwards recursive formula for beta. ~~As the~~ The sp1p2 data last read in local memory 20 is read into register 34, which contains not only the sp1p2 data, but the extrinsic value (which in the initial stage operation is 0), as well as the data representing the memory location in which the initial extrinsic value located in the extrinsic memory 14. The sp1p2 an extrinsic data undergo calculation at gamma calculation stage 36. The output of the gamma calculation stage 36 is applied to gamma registers 38 and 40. The beta calculations are respectively performed by beta calculation stage 44 and beta normalization stage 46. Initially, a start condition of binary one ("1") is applied to input 42a of multiplexer 42. The normalized beta calculation is initially applied to extrinsic value calculation stage 50 through output register 48 which further applies the last calculated input to input 42b of multiplexer 42. Extrinsic value calculator stage 50 calculates an extrinsic value for each time state k by looking at the alpha value for register 52 received at input ~~58~~ 50a, the gamma value from register 38 received at input 50b and the beta output from register 48 received at input 50c. Registers 48, 52 and 38 are provided to assure time registration of the signals at the extrinsic value calculator stage 50.

[0034]   As was described hereinabove, when an extrinsic value has been calculated, i.e., upon the completion of the first ~~interration~~ <u>iteration</u>, this extrinsic value is read out of extrinsic memory 14 and is used during the calculations of the next iteration. Conventional control circuitry, not shown for purposes of simplicity, determines the number of iterations to be performed.

[0036]   Although the embodiment shown in Figure 3 teaches an implementation in which alpha is calculated during the first window and beta is calculated during the latter portion of the first window, it should be understood that the alpha and beta calculations may be reversed, as shown in Figure ~~5~~ <u>3a</u>, while still deriving all the benefits of the embodiment shown in Figure 1, namely the significantly reduction in calculation time as well as a 50% reduction in the memory requirements for the turbo decoder of Figure 3 as compared with present day techniques and apparatus.   The architecture of the present invention enables further reductions in memory size. For example, the data may be processed using three (3) windows, four (4) windows, etc, which provide further reductions in memory size. For example, using four (4) windows results in a memory size that is 3 the memory size compared to processing where no windowing is employed.

[0038]   Although the above-described architecture was developed for a turbo decoder, all convolution codes can use an MAP decoder. The calculation of the forward metrics may be calculated before or after the reverse metrics. The reverse

metrics could be calculated first, and then the forward metrics can be calculated while the output calculations are performed. This can be accomplished as shown, for example, by the embodiment of Figure 3a wherein calculation block 24^1 is a beta calculator; calculation block ~~24~~ <u>26'</u> is a beta normalization calculation block, memory 32^1 is a beta memory; calculation block 44^1 is an alpha calculation block and calculation block 46^1 is an alpha normalization calculation block.